DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A DECISION MODEL FOR SELECTION OF MICROCOMPUTERS AND OPERATING SYSTEMS

by

Keith G. Highfill

June 1984

Thesis Advisor: Norman Schneidewind

Approved for public release; distribution unlimited

T217408

[illegible]
NAVAL POST
MONTEREY, CALIFORNIA

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A Decision Model for Selection of Microcomputers and Operating Systems		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Keith G. Highfill		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE June 1984
		13. NUMBER OF PAGES 75
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Microcomputers, decision support models, systems analysis, economic analysis, operating systems

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A framework for the construction of an economic analysis model is suggested for the selection of microcomputer hardware and operating systems. The model is suggested in order to guide a prospective Navy microcomputer purchase through the large number of available microcomputer systems. The model is designed such that common "spreadsheet" software programs can be utilized to manipulate the model and store data on available systems. (Cont)

ABSTRACT (Continued)

In addition, comparisons are made of current popular microcomputer systems and operating systems, in order to provide a frame or reference for the use of the model.

Approved for public release; distribution unlimited.

A Decision Model for Selection of Microcomputers and Operating Systems

by

Keith G. Highfill
Lieutenant, United States Navy
B.A., University of Rochester, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
June 1984

ABSTRACT

A framework for the construction of an economic analysis model is suggested for the selection of microcomputer hardware and operating systems. The model is suggested in order to guide a prospective Navy microcomputer purchaser through the large number of available microcomputer systems. The model is designed such that common "spreadsheet" software programs can be utilized to manipulate the model and store data on available systems.

In addition, comparisons are made of current popular microcomputer systems and operating systems, in order to provide a frame of reference for the use of the model.

TABLE OF CONTENTS

I. INTRODUCTION 9
   A. BRIEF MICROCOMPUTER HISTORY 9
   B. THE MICROCOMPUTER MARKET 10
   C. GOALS OF THESIS 11

II. BACKGROUND 13
   A. MICROCOMPUTERS IN THE NAVY 13
   B. MICROCOMPUTERS AND SYSTEMS ANALYSIS 16
   C. METHOD AND ASSUMPTIONS 18

III. DECISION MODEL FRAMEWORK DESCRIPTION 24
   A. MODEL ENVIRONMENT 24
   B. MODEL DESCRIPTION 25
   C. EXAMPLE ANALYSIS 28
   D. MODEL ENHANCEMENTS 35

IV. GENERAL MICROCOMPUTER DESCRIPTION 37
   A. BASIC CONCEPTS 37
   B. THE MICROPROCESSOR 46
   C. MICROCOMPUTER SYSTEM CONSIDERATIONS 48

V. CURRENT 16-BIT MICROPROCESSOR DESCRIPTIONS 51
   A. INTEL 8086/8088 MICROPROCESSORS 51
   B. ZILOG Z8000 MICROPROCESSOR 55
   C. THE MOTOROLA MC68000 MICROPROCESSOR 57

VI. EXAMPLE 16-BIT COMMERCIAL OPERATING SYSTEMS . . . 61
   A. DIGITAL RESEARCH CP/M-86 61
   B. MICROSOFT MS-DOS AND IBM PC-DOS 65

VII. CONCLUSIONS AND RECOMMENDATIONS 69

LIST OF REFERENCES 71

BIBLIOGRAPHY 72

INITIAL DISTRIBUTION LIST 75

LIST OF TABLES

I. BCR Comparison of Alternatives 27
II. Graphic Display Options 28
III. Hard Disk Options 31
IV. RAM Expansion Options 33

LIST OF FIGURES

2.1 Economic Analysis Process 19

3.1 Graphic Display Options Graph 29

3.2 Hard Disk Storage Capacity Graph 32

3.3 Additional RAM Expansion Graph 34

I. INTRODUCTION

A. BRIEF MICROCOMPUTER HISTORY

The microcomputer market is generally believed to have begun when the Micro Instruments and Telemetry Systems (MITS) Altair 8800 computer kit was introduced in late 1974. MITS expected to sell 200 or 300 of the computer kits in 1975. Instead they were deluged with orders for over 2000, selling over 200 in one afternoon alone. In 1977 Apple Computer Inc., began shipping its Apple II microcomputer and promptly began growing at a rate of more than seventy percent annually. The market demand for microcomputers is still being underestimated in 1984. When International Business Machines (IBM) introduced its Personal Computer (PC) in August of 1981 few could have predicted the phenomenal demand for the product. It has been estimated that over one million IBM PC's were shipped in 1983 and that IBM expects to ship two million in 1984. Despite the volume many customers must wait months to receive their orders and the shortage has created a large market for IBM compatible microcomputers.

One of the reasons for the excessive demand is that no one realized that large and small corporations would buy the IBM PC and other microcomputers in volume. One example is Travelers Insurance Co., which purchased 2000 IBM PC's and has indicated with a letter of intent to buy 10,000 more [Ref. 1]. Universities are another large buyer of personal microcomputers and in particular the IBM Personal Computer. The University of Waterloo, popular for its teaching and learning compilers, purchased over 300 PC's to shift undergraduate programming and engineering courses off their IBM

3033 mainframe to the PC's. Almost weekly there is a new announcement of another college or university that will require all incoming freshman to purchase a particular school endorsed microcomputer. Even graduate schools have joined the trend with Harvard University announcing that all entrants to its business school will be required to purchase IBM Personal Computers.

B. THE MICROCOMPUTER MARKET

Microcomputers now account for 32.3 percent of the total available computer product market, and is growing at a rate of 26.9 percent per year. This is compared to a 5.5 percent growth seen in the mainframe and supercomputer market. There are now over 500 companies manufacturing microcomputers. These companies combine to form over 950 different models of microcomputers from which to choose [Ref. 2].

The cover story for the August 8, 1983 issue of Business Week [Ref. 3] is entitled "Computer Shock Hits The Office: A Wild Proliferation of Desktop Units is Confusing Everyone". The article is accompanied by a cartoon that shows a customer standing in a computer store with rows and rows of microcomputers asking the sales clerk, "Do you have one that can help me decide which is the one for me?". Certainly the personal computer and microcomputer market is experiencing what many have termed explosive growth. Joseph Ferreira, a vice-president for Diebold Group Inc., in the article states that executives are suffering from option shock and cannot handle all the choices [of microcomputers] coming to them. Later Ferreira also states that they will "suffer paralysis by analysis", implying that the choices are so many that a consensus decision cannot be reached. With over 950 models of microcomputer systems available for commercial and consumer use it is easy to understand how someone given the task of selecting the best system for a particular application might consider the task near impossible.

C. GOALS OF THESIS

The Navy purchaser is therefore left with a complex maze of choices for microcomputer purchase options. As the overall cost of microcomputer systems continues to fall, the individual incentive to perform a proper and accurate systems and cost analysis for a single and isolated microcomputer purchase decreases. This is not to say however that a complete needs and economic analysis should not be performed, rather the Navy purchaser must be made aware of the overall consequences of not performing the proper analysis. While the isolated purchase of a single microcomputer system may seem inconsequential, when viewed and totaled over all Navy personnel making such decisions, there exists an enormous potential for waste of valuable Navy dollars. Further, and more fundamental, the Navy purchaser may acquire a system that will not satisfy user needs.

It is therefore desirable to provide an effective decision support model and criteria to aid the Navy purchaser in the selection of microcomputer systems. This thesis will develop and describe a conceptual framework from which it is hoped that an effective decision support model can be constructed for selection of microcomputer systems. It is hoped that the model design can be implemented on common "spreadsheet" software programs to allow manipulation of the model and storage of data on available systems.

The proposed model is intended to apply only to single microcomputer purchases, and be only part of a complete system and cost analysis. It is felt that automating as much of the analysis process as possible not only reduces the time and cost of conducting the analysis, but produces an incentive for Navy decision makers to conduct a more complete analysis. Further, and in answer to the cartoon in Business Week, microcomputers can be just the tool to aid the analysis process.

II. BACKGROUND

A. MICROCOMPUTERS IN THE NAVY

Within the Navy, the Naval Data Automation Command (NAVDAC) is assigned overall responsibility for microcomputer support. NAVDAC has assigned primary responsibility for the development of microcomputer technology and its applications to Navy Regional Data Automation Center (NARDAC) Norfolk, Virginia [Ref. 4]. NARDAC Norfolk is tasked with the following microcomputer support functions:

1. Develop standards for microcomputers, including interfacing with mini and large host computers.
2. Develop a standard Request for Proposal (RFP) for competitive procurement leading to a contract for microcomputers, associated maintenance support, and licensing of software for Navy-wide use.
3. Develop microcomputer guidelines on procurement, system software, hardware and application program selection.
4. Develop an inventory of Navy owned software, and listings of available vendor hardware, software, and training.
5. Develop microcomputer training courses to be given by NARDACs and Naval Data Automation Facilities (NAVDAFs) in support of area activities.
6. Host workshops on microcomputers for Navy-wide users.

In order to guide the prospective Navy microcomputer purchaser NAVDAC developed guidelines for equipment and operating systems for 8-bit microcomputers [Ref. 5]. Microcomputer technology however did not allow 8-bit microcomputers to remain entrenched for long. NAVDAC's initial response to 16-bit microcomputers was that most users probably did not need a 16-bit microcomputer system and continued to refer users to the recommended guidelines for 8-bit systems. As 16-bit systems became increasingly more popular it was realized that standards should also be developed for 16-bit systems. In the interim NAVDAC recommended that 16-bit systems be purchased as a complete microcomputer system including hardware and software from a single vendor. This was to insure that all components worked together and to provide a single point for vendor support.

It was not long before NAVDAC decided that 16-bit systems had matured to a point where they no longer recommended microcomputer systems based on 8-bit technology. The 16-bit microcomputer systems were recognized to have superior capabilities and the sophisticated software developed for these systems were addressing more user needs, and running existing applications quicker and more efficiently. For 16-bit microcomputers a preliminary decision was made to not specify hardware requirements but to recommend compatibility with a particular operating system only. This is in light of increasing numbers of microcomputers becoming "compatible" with each other. The initial operating system chosen was Microsoft's Disk Operating System (MS-DOS) due to its wide acceptance as a de facto standard. Over 96 percent of the 16-bit microcomputers use either MS-DOS or IBM's PC-DOS which is essentially equivalent to MS-DOS.

The popularity of the IBM Personal Computer and compatible machines is having a significant impact on the microcomputer market. This impact is resulting in application software (for microcomputers) being developed almost exclusively for the IBM PC and compatible systems. NAVDAC felt that some level of IBM PC compatibility was therefore necessary in establishing 16-bit standards. The guidelines for 16-bit microcomputers have been drafted and are awaiting approval. These guidelines as developed by NAVDAC require operating system and media[1] (data) compatibility with the IBM PC. Further, the draft guidelines will replace the 8-bit hardware and software standards and 8-bit microcomputer systems will no longer be recommended.

NAVDAC Advisory Bulletin No. 54 [Ref. 6] announced the award of a joint Air Force/Navy contract with Zenith Data Systems for the delivery of up to 500 systems per month of the Zenith Z-120 Microcomputer system. The Zenith Z-120 Microcomputer is not program compatible with the IBM PC. In other words, a program written for the IBM PC will not necessarily run on the Z-120. The two systems are media compatible and share almost identical operating systems. NAVDAC therefore faced the possibility of recommending guidelines that were not compatible with the chosen system. Therefore data compatibility is the only criteria for the NAVDAC 16-bit guidelines.

The joint Air Force/Navy contract allows the prospective Navy purchaser of a 16-bit microcomputer to obtain the Zenith Z-120 at a substantial discount from the going market retail price. The user base for the Zenith Z-120 in the Navy is therefore expected to be large. There are however large numbers of IBM PC and compatible machines already purchased and in use. In addition, although there is an increasing number of software packages available for the Zenith Z-120, there are thousands of software packages written for the IBM PC and other compatible systems that will not operate on the Zenith Z-120 because the systems are not program compatible.

---

[1] Media compatibility in this context means that data is stored on equivalent external storage devices in the same physical format and thereby allowing the data produced by one system to be read or accessed by the compatible system.

Further, media compatibility does not necessarily mean that the data produced can be used by another system. An example of this is an application program that indexes or perhaps hashes its data elements in order to increase access speed and minimize storage space. This imposes a certain logical structure on the data elements. Another system that is only media compatible may be able to physically read the data disk, but the data will appear as a random stream of bytes with no useful meaning. Only the same application program or one especially developed for that data can decode the logical data formats to a useful, readable or original form.

B. MICROCOMPUTERS AND SYSTEMS ANALYSIS

It should be emphasized that standard systems analysis must be performed when possible. If anything, the large number of microcomputers in varying degrees of compatibility make systems analysis even more important than before. NAVDAC guidance and standardization efforts go a long way to eliminate some of the difficulty, but since no single machine or hardware standards are exclusively recommended, not all of the purchasing problems are solved. Therefore, the need still exists to perform systems decisions, and the criteria of data compatibility only limits the field of choices available. Further, as 16-bit microcomputers continue to evolve, a method is needed to easily evaluate the changes and their effect on the existing recommended standards.

Any proposed model for evaluating microcomputer systems should fit into standard systems analysis techniques. In those cases where a full systems analysis is not performed, the method should provide a useful and accurate tool for microcomputer selection consistent with standard systems analysis goals. One of those goals is to acquire the "best" system from among the chosen alternatives at least cost.

Traditional systems analysis can be broken down into four broad categories in a system life cycle concept [Ref. 7].

They are:

1. The Study phase.
2. The design phase.
3. The development phase.
4. The operation phase.

The prospective Navy purchaser of microcomputers however, when faced with a buying decision, does not often see the purchase of a microcomputer as a part of a particular larger information system. The microcomputer is seen as the entire system, where no design or development is required. Design and development have already been accomplished. The operation phase is reduced to merely purchasing an extended warranty contract, where no real or significant operating costs are seen or envisioned. The prospective microcomputer purchaser therefore sees their role limited to the study phase. That study phase is limited to selecting which microcomputers from among many will be purchased.

The above assumptions attempt to point out the realities of the purchasing situation as opposed to what properly should be done. NAVDAC cautions the prospective purchaser [Ref. 5] that indeed the first and most important consideration is to perform a proper activity requirements and needs analysis to economically justify the purchase of microcomputer systems. Further, the microcomputer should be considered as just one part of an information system with particular hardware, software, personnel, and facilities support requirements and evaluated within a full system life cycle concept.

NAVDAC PUB. 15 [Ref. 8] provides guidance and procedures on how to conduct economic analysis. Particularly emphasized are the procedures for performing present value analysis and discounted payback analysis. NAVDAC PUB. 15 is oriented towards large automatic data processing (ADP) system decisions. As such it provides analysis techniques that evaluate among widely differing alternatives such as whether to install or change the current ADP system in-house or utilize an outside service organization ADP system.

The techniques provided by NAVDAC PUB. 15 can be applied to microcomputer systems. Figure 2.1 from NAVDAC PUB. 15 will be used as the basis for describing how the proposed model fits into the economic analysis process.

C. METHOD AND ASSUMPTIONS

The proposed decision support model is a method to be utilized for evaluating the differences among many microcomputer systems. The differences can be among several configurations of the same system or between several separate systems. The method utilized will be to quantify system attributes and parameters where possible and formulate a model that allows comparisons of the system attributes and parameters against cost data.

The model forms its comparisons by first calculating cost/benefit ratios for each of the given parameters or attributes. The model then leads the user through comparisons between the alternative systems or configurations under consideration. The comparisons are calculations in the changes between benefits and costs of the alternatives. The key to the model is the calculation of delta's, or the changes between benefits and costs. It will be shown that the calculation of cost/benefit ratios alone as per NAVDAC 15 is not sufficient. Rather it is the change or delta

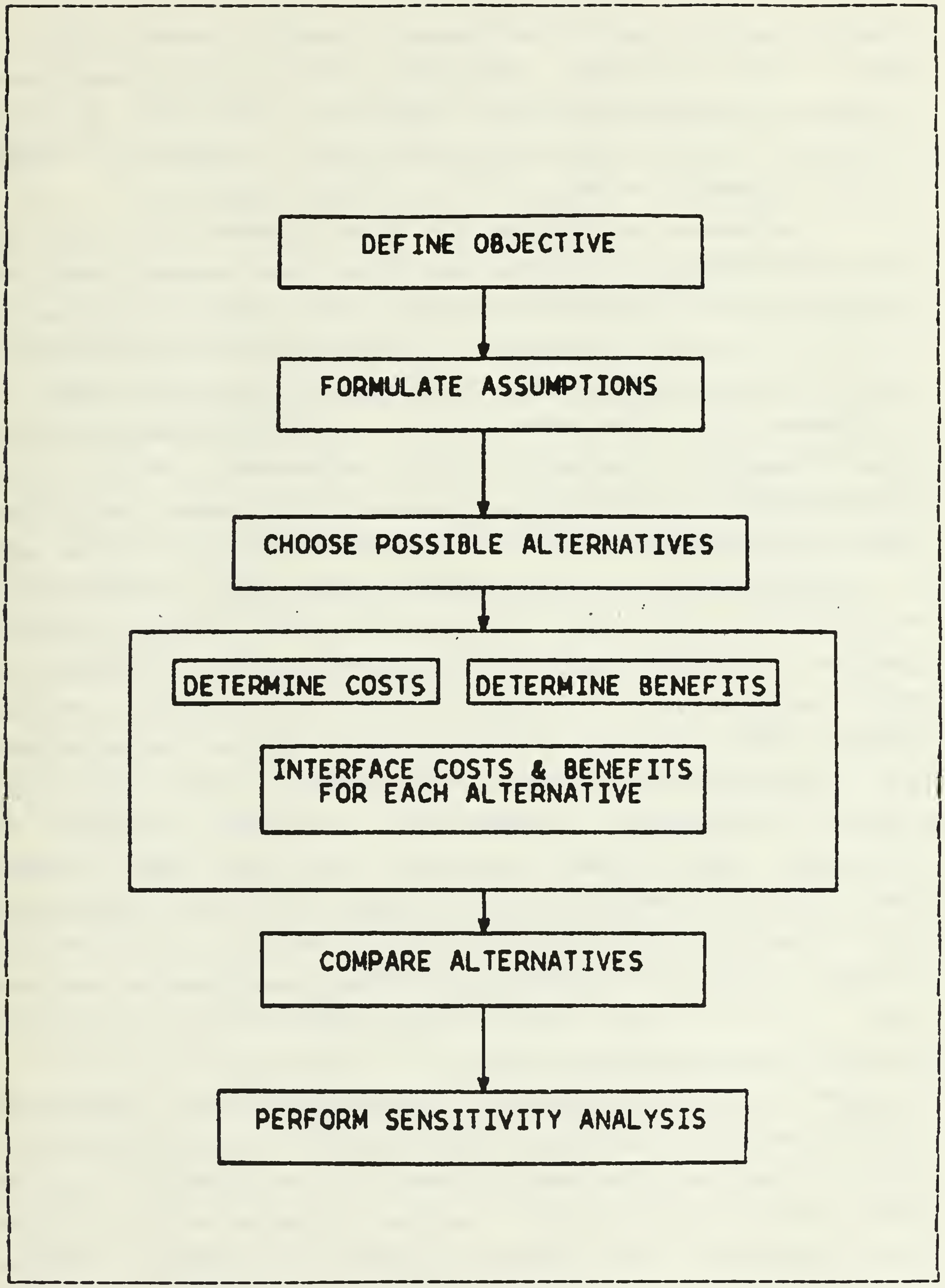
DEFINE OBJECTIVE
FORMULATE ASSUMPTIONS
CHOOSE POSSIBLE ALTERNATIVES
DETERMINE COSTS
DETERMINE BENEFITS
INTERFACE COSTS & BENEFITS FOR EACH ALTERNATIVE
COMPARE ALTERNATIVES
PERFORM SENSITIVITY ANALYSIS

Figure 2.1 Economic Analysis Process.

value (change in benefits/change in costs) between alternatives that should be among the factors considered when comparing alternatives. The model is a tool and is merely designed to perform the calculations, it does not necessarily recommend or determine the "best" system.

In reference to figure 2.1 it is assumed that the user has defined his objectives, formulated the assumptions, and chosen the possible alternatives. The model performs the task of interfacing the costs and benefits for each alternative, and therefore assumes that costs and benefit data can be determined. The model then displays the resultant data and calculations in a form that allows the comparison of alternatives. The model is designed to be used for comparison of microcomputer system attributes and parameters only.

Again it should be emphasized that this process should normally be only part of the complete system analysis process. It assumes that the need for a microcomputer system has already been justified and established. Further, commands must realize that the cost of the microcomputer itself will not be the only cost in the life-cycle. Application programs, update fees, accessory equipment and supplies, maintenance contracts and for some commands training of personnel are just some of the additional costs that should be considered. These costs are separate from the cost of the microcomputer after a selection has been made, and in some instances may provide the overriding factors in the decision.

The use of the proposed model alone would assume that the follow-on costs are basically equal, and this is not a valid assumption for many systems. It is hoped that the model however will provide a means to choose among microcomputer systems when complete systems analysis is not performed and all of the significant costs are considered. Furthermore, it is hoped that it will be an excellent addition to the complete systems analysis process.

Most often system comparisons of similar products is performed using matrix parameter tables. These tables usually list the particular systems along one axis and system attributes along the other axis. These tables are often large and confusing with many attributes listed. Often a particular system is just represented by a check mark or bullet point if it possesses a certain attribute, better evaluations may give more descriptive or amplifying information for the appropriate entry.

More often than not the attributes listed are parameters that can be quantified. One of the basic tenets of systems and economic analysis is to quantify attributes where possible. Once quantified, the difficult-to-analyze matrix can give way to useful figures that not only compare alternatives but also indicate the relative advantages of one system over another.

Further, the requirement of finding the data to be quantified is often difficult in itself. This is where a simple but effective model is useful. It is hoped that the data required to develop the model can be easily obtained. This is normally true for microcomputer systems. If they required data cannot be obtained to generate the model, then obviously the model loses its viability. The model should therefore be as simple as possible, but care must be taken to ensure that the data and parameters chosen are complete enough to make the model valid.

Another key aspect of comparing alternatives is that there must be a baseline from which to compare. Normally the baseline is to be developed from the current system in use. For the Navy purchaser of microcomputers a current system may not be in place. The baseline can then be derived in two ways. First, would be to use the existing 8-bit or 16-bit standards and guidelines as developed by NAVDAC. Second, and more preferable, would be to use a baseline developed from user requirements as much as possible. This would assume that the prospective purchaser is able to quantify requirements in context to the model developed. If a current microcomputer system is in place, then that system can be used as the baseline.

Care must be taken not to confuse baselines with weighting factors. Weighting factors are criticized as being arbitrary and often chosen to favor particular systems. The baseline does not modify the parameters but provides only a figure for comparison. This allows the figure to show how the proposed system compares relative to the existing system or the developed NAVDAC standards. The proposed model utilizes no weighting factors and modifying the model with weighting factors is not recommended.

The proposed model evaluates single user, single task microcomputer systems. Current microcomputer technology has reached the point where multiple users and multiple tasks can be performed. Extensions to the model to cover multiple users and multiple tasks must be developed when these microcomputer systems become more commercially available and are under consideration.

Another criticism of any quantification scheme is the argument over the validity and usefulness of the chosen parameters. This model design intentionally attempts to keep the chosen parameters as simple as possible. This is done for two reasons. First, is to ensure that the parameter data can easily be obtained by the Navy purchaser. Second, the model must be easily understood so that people will use it. Complex simulations, queuing models, benchmark tests, and other methods are all important tools, but are useless to the average Navy purchaser of a microcomputer system who has neither access to the required data nor the understanding to effectively utilize them. Further, a simple model may lead to the same conclusion as those performed by complex analysis.

The Army/Navy Computer Family Architecture Committee [Ref. 9] and [Ref. 10] performed a study on nine candidate computer systems for possible selection for a future ruggedized version for military applications. Only three systems were acceptable from what was defined as an absolute criteria standpoint. Further analysis was performed on qualitative criteria and results were that the same three systems came out on top. Exhaustive benchmarks, program performance and statistical design tests were performed and resulted in the same relative rankings as resulted from the qualitative criteria.

The proposed model is not intended to be static, but rather a dynamic device to be utilized as necessary by the prospective purchaser. The prospective Navy purchaser is encouraged to add or delete chosen parameters when more detailed parameters are desired or certain parameters cannot be obtained.

III. DECISION MODEL FRAMEWORK DESCRIPTION

A. MODEL ENVIRONMENT

The proposed model is intended to be implemented on any of the common "spreadsheet" type programs. As such, no specific details of implementation or how to construct the model are described. The methods discussed are merely intended to provide a framework from which it is hoped that an actual complete working model can be constructed. The calculations and data manipulation performed by the model are all easily within the capabilities of any of the popular spreadsheet programs.

A spreadsheet program merely serves as a very valuable tool to help analyze model data; it is not required. All of the calculations and analysis can be performed by hand. It is felt however, that the automation of as much of the analysis as possible will provide incentive to perform the analysis by removing the time required to perform numerous detailed calculations by hand. Further, an automated spreadsheet database once constructed can be used by many and the changing and updating of the required data is easily performed.

Spreadsheet programs that offer two additional capabilities would provide even more utility. The capabilities are graphing and the ability to construct macros. Graphing allows the data to be viewed in a more beneficial visual form. The ability to process macros allows the model to begin to perform as an elementary decision support system by allowing the model construction and operation to be menu driven and user prompted. There are a number of advanced spreadsheet programs that offer these additional capabilities, particularly for 16-bit microcomputers.

B. MODEL DESCRIPTION

The model consists of tables of cost and benefit information. The benefits consist of various microcomputer system technical parameters that have measurable and comparative attributes. The technical parameters or performance attributes are being used as surrogates for actual user benefits, this is the best the user can do lacking detailed information and measurement of the actual benefit to the user. Examples would be size of system random access memory (RAM) or speed of system operation in megahertz. Chapter 4 discusses various system parameters that could be utilized, and chapter 5 and 6 detail these parameters for current components and systems. The costs should be the specific cost of the particular attribute involved. For example, if secondary disk storage devices are being compared, the cost of the devices alone should be entered and not the total system cost.

There will obviously be parameters where separate cost data cannot be obtained. In these cases the cost data can utilize total system cost when the configurations are identical except for an attribute of the of the user's requirement. The analyst must be extremely careful however that alternative systems costs are associated with identical configurations, i.e. the alternate systems in that particular comparison should equally satisfy user requirements. If the configurations are not identical, then the differences in benefits must be among parameters whose differences are not considered important to the decision process.

An example of this would be comparing two systems that offer different display screen resolutions where no separate costs can be identified, but one system has color capability and the other does not. Screen resolution could be a parameter benefit and compared against total system cost only if color capability is not a system requirement and not considered a decision parameter by the user. If separate costs cannot be attributed to particular parameters, then this model cannot be used for cost benefit analysis of those parameters.

After the system costs and benefits have been entered for all technical attributes relevant to user needs, then the model calculates benefit/cost ratios for all alternatives. Benefit/Cost Ratio (BCR) is defined as the quantifiable parameter measure divided by the cost of that parameter. This assumes that the associated costs are one time initial costs. NAVDAC Pub. 15 provides techniques to calculate uniform annual costs (UAC) and net present values (NPV) if required. The benefit/cost ratios are then displayed and maintained for reference purposes.

The model then calculates the delta or change values in the benefits versus the costs for pairs of alternatives under consideration. Depending upon the specific model construction and spreadsheet program in use, this can be automatically calculated for all opposing sets of alternatives or calculated as a result of user prompts. Normally, the delta (change) values should be calculated between the known baseline, and various alternatives (If there are many alternatives, pair-wise calculations will rise exponentially).

As stated previously, the baseline can either be developed from user needs and specifications, NAVDAC current standards, or against a current system in operation. The delta calculations then show the highest benefit-cost difference relative to the selected alternative or baseline.

The individual analysis (comparison of alternatives) and conclusions drawn from the benefit/cost ratios is then left to the analyst. NAVDAC Pub. 15 provides guidance on the use of benefit/cost ratios. According to this publication, benefit/cost ratios are normally utilized when the alternative systems have unequal costs and unequal benefits. This is the most frequent case at present when trying to evaluate the differences among many similar microcomputer systems, when one system has some attribute slightly different than the others. NAVDAC Pub. 15 provides guidance on conclusions to be drawn from benefit/cost ratio analysis.

TABLE I

BCR Comparison of Alternatives

Costs	Benefits	Basis for Recommendation
Equal	Unequal	Most Benefits
Unequal	Equal	Least Costs
Unequal	Unequal	Highest benefit/cost ratio
Equal	Equal	Other Factors

It will be shown that when comparing alternatives, the analyst must look at both the delta values for the BCR and the individual BCR's themselves. This is the key point to this model construction and method. Further, the key aspect to examining the delta values is to begin with the least cost alternative that satisfies user requirements and examine the delta values increment by increment progressing from one alternative to another. Many alternatives can be eliminated by bounding the cost and attribute data by maximum available funds willing to spend for cost, and minimum acceptable measure of a particular attribute. The analysis then shows how much incremental cost is required for some increment in performance. A decision is reached when the incremental cost is not considered worth the incremental benefit or vice-versa.

No attempt will be made to list or detail all the system attributes that should be considered in the analysis, or how the cost/benefit ratios and the changes in cost benefit ratios should be calculated for each attribute. Rather the intent is to make the prospective Navy purchaser aware of the technique of conducting change analysis, and suggesting that automated spreadsheet tools provide the ideal environment for conducting not only cost/benefit analysis, but the entire economic analysis process.

In order to guide a prospective builder of an actual working model, several model examples will be discussed.

C. EXAMPLE ANALYSIS

Assume that user requirements have specified a need to display graphics. The model should scan those systems entered into its database and select those systems with graphics capabilities as the alternatives. Table II is displayed for consideration and Figure 3.1 is made available

TABLE II

Graphic Display Options

Vendor	Pixel Resolution	Total Pixels	Cost	Total BCR
A	640x225	144000	350	411.42
B	640x200	128000	400	320
C	600x200	120000	480	250
D	640x400	256000	480	533.33
E	720x348	250560	500	501.12
F	720x400	288000	540	533.33

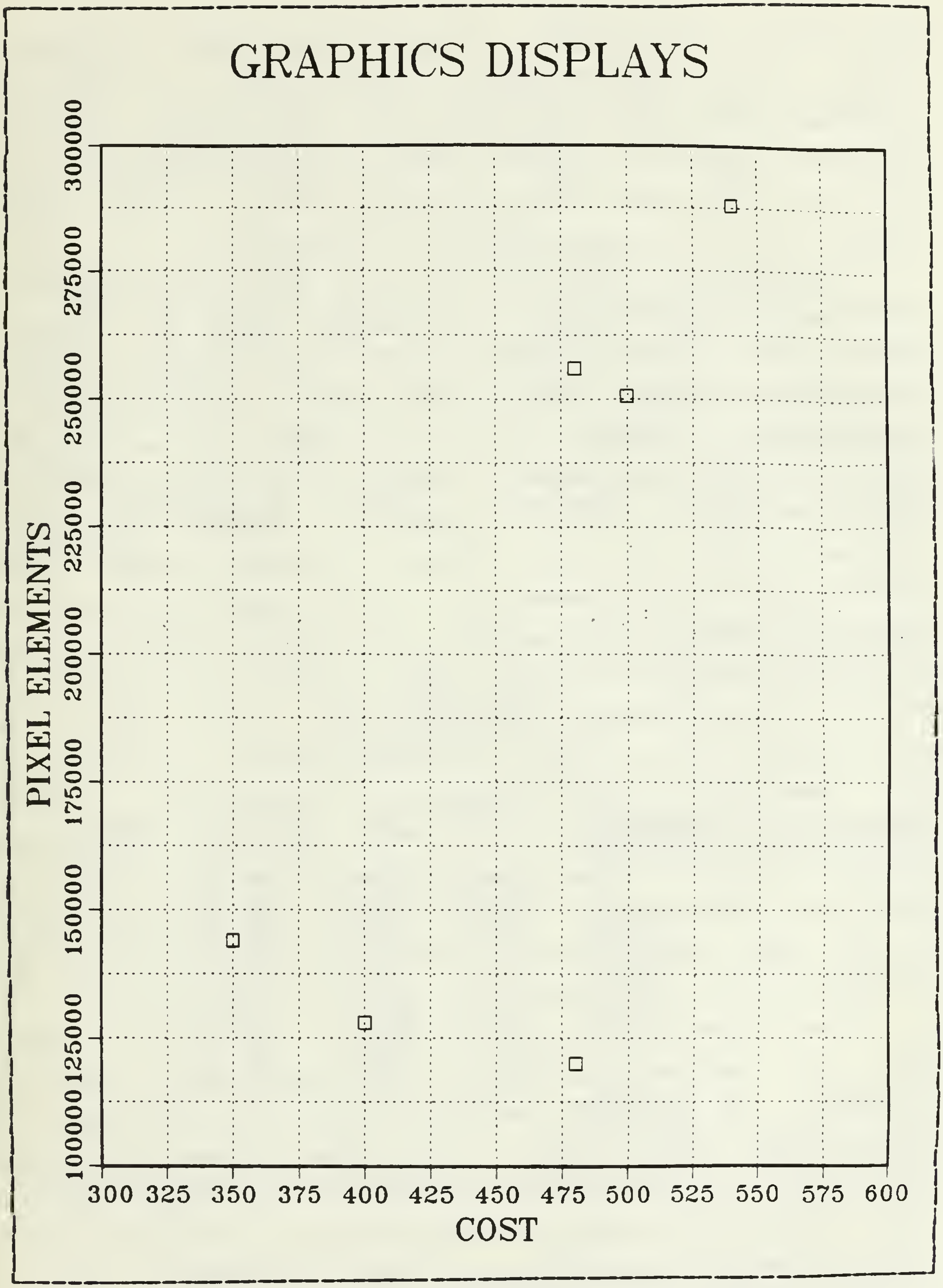

Figure 3.1 Graphic Display Options Graph.

for viewing. The data necessary to generate Table II and graph 3.1 represents actual cost data for six different manufacturers of graphic display options for the IBM Personal Computer. Vendor A offers a graphic screen resolution equal to that of the Zenith Z-120 adopted by NAVDAC. Vendor A will therefore represent the baseline.

Examining the BCR's and following the NAVDAC guidelines, the analyst might be drawn to recommend Vendor F. Vendor F and Vendor D have the highest BCR's, but Vendor F offers a considerably higher vertical resolution. Assuming that all other factors were the same, this would be an erroneous conclusion for two reasons. First, the conclusion is not based on need. Vendor A should be selected because it represents the least cost solution, assuming Vendor A would satisfy the user requirement. In this case even though the costs and benefits are unequal, the analyst must be careful to evaluate unequal benefits first against the stated need. If all the alternatives equally satisfy the user need, then in actuality, all of the alternative benefits are equal, and the least cost alternative should be selected. Second, assuming that Vendors D and F were the only two alternatives, it must be remembered that even though the costs and attributes are unequal, if the systems are of equal benefit to the user, and the BCR's are equal,then the least cost recommendation criterion should apply.

If we assume however that Vendor A represents the baseline of a current system, or the user need specifies exceeding the baseline parameter, then an evaluation of delta values must be done. The model or user would then calculate the delta functions between the baseline and the other alternatives.Delata values should be calculated in all cases where actual benefits and costs are unequal.

When Vendor A is compared to Vendor D the change in pixels equals 112,000 and the change in cost equals 130, yielding a delta BCR of 861.54. When Vendor A is compared to Vendor F, the change in pixels equals 144,000 and the change in cost equals 190, yielding a delta BCR of 757.89. From the change analysis the analyst should be led to recommend Vendor D, which offers the most extra benefit at least extra cost. If possible the display of the delta BCR's should also be shown in a table for all alternative pairs.

The next example will examine differences among manufacturers of Hard Disk Drive options for microcomputer systems. Again, the data represent actual cost data for manufacturers of hard disk options for IBM compatible microcomputer systems. Table III and Graph 3.2 would be displayed by the model calculations. Vendor D represents the baseline NAVDAC standard.

TABLE III

Hard Disk Options

Vendor	Capacity	Cost	Total BCR	Delta BCR (Relative to Baseline)
A	5 MB	1540	.0032	--
	10 MB	1840	.0054	--
	15 MB	2310	.0065	.0056
	20 MB	2550	.0078	.0089
B	12 MB	2180	.0055	.0026
	20 MB	2440	.0082	.0098
	26 MB	2630	.0099	.0132
C	10 MB	1495	.0067	--
	23 MB	2250	.0102	.0157
D	10 MB	1422	.0070	Baseline

As in the first example, the least cost system is represented by the NAVDAC standard. This alternative should be

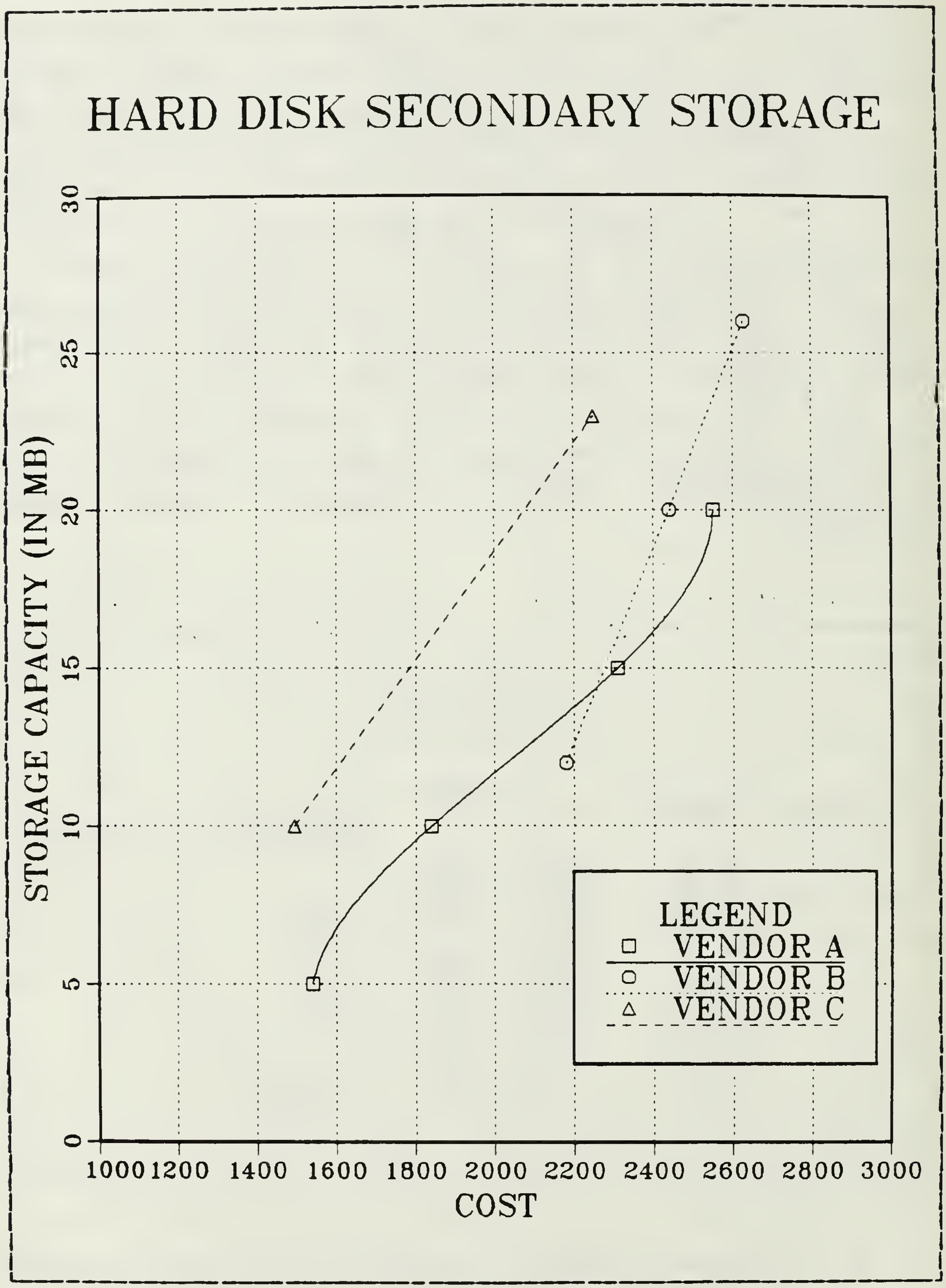

Figure 3.2 Hard Disk Storage Capacity Graph.

selected unless user requirements specify a need for secondary storage capacity greater than 10 Megabytes. Assuming greater secondary storage is required than the amount offered by Vendor D, then the analysis data indicates that Vendor C's 23 MB unit offers both the highest BCR and delta BCR from the established baseline.

The last example examines the analysis of different RAM expansion options for a single system. Assume an existing system already has 256 kilobytes of user RAM installed. Two vendors manufacture memory expansion options. Vendor B's product only has the capability to expand to 256 KB (for 512 KB total); Vendor A can expand an additional 384 KB (for 640 KB total), but at additional cost. Table IV and Figure 3.3 represent possible data displays.

TABLE IV

RAM Expansion Options

Total Ram	256	320	384	448	512	576	640
Vendor A Cost	295	350	405	460	515	570	625
Total BCR	.868	.914	.948	.974	.994	1.01	1.024
Vendor B Cost	229	284	339	394	449		
Total BCR	1.12	1.13	1.13	1.14	1.14		

The cost data for the two Vendors is also taken from actual memory expansion options for IBM Personal Computers. Memory add-on boards are an extremely popular option, and the graph and table represent a common occurrence where one vendor offers more expansion capability but at a higher price. Either expansion board can be filled with memory at equal additional cost. The table shows that traditional

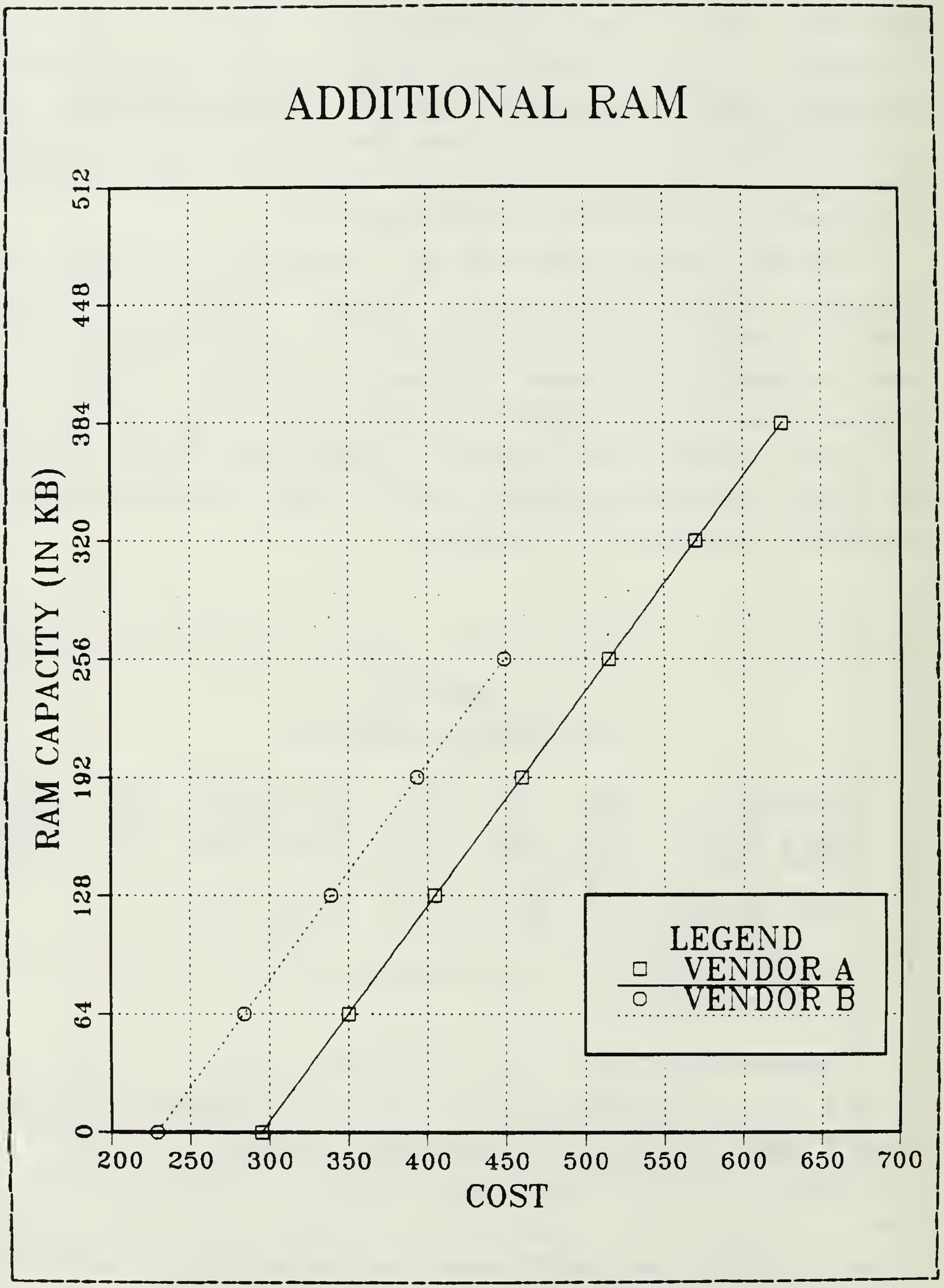

Figure 3.3 Additional RAM Expansion Graph.

benefit cost ratios indicate the lowest price vendor (B) should be recommended throughout the range of the expansion options.

There are several different change functions that can be calculated for this case. If the model calculates the difference in expansion capability based on the initial cost; then Vendor A can expand an additional 384 KB at a cost of 295, giving a delta BCR of 1.30. Vendor B can expand an additional 256 KB at an initial cost of 229 yielding a delta BCR of 1.12. The model could also calculate the difference at full cost including cost of memory. Vendor A at the 384 KB change, full memory cost is 625; Vendor B with 256 KB change, at full memory cost is 449. This yields delta BCR's of .6144 and .570 respectfully. Thus the BCR when measured as the difference between expansion capability from the system initial capability indicates that Vendor A, the higher priced vendor, should be recommended, assuming the user requires this added capacity.

It must be emphasized again that the above analysis is not based on a user need requirement. Only when the actual need is considered can actual recommendations be made. For the above example, expansion of memory capacity beyond what was originally a maximum requirement occurs frequently, but if it is determined that 256 KB of memory is the absolute maximum requirement, then the least cost option should be selected.

D. MODEL ENHANCEMENTS

The above three examples are representative of how benefit/cost analysis can be performed in reference to comparing alternatives among microcomputer systems. An actual working model would be constructed based on the individual decision makers or analysts requirements. These requirements should be quantification of measurable system attributes based on user need. As such, many different parameters would have to be compared. Ideally, a constructed model would be preloaded with attribute and cost data on all known alternative systems. The user would then extract those parameters necessary, and conduct the individual analysis on each of the chosen parameters.

In the realm of decision support system theory, the model could be just one of a series of models used in making microcomputer selection. If a rule dictionary could be developed for user requirements an ideal decision support system could be constructed that would automatically select the appropriate evaluation parameters based on the stated user need. The cost/benefit analysis would be just one of several models that a prospective user is guided through in the decision process. Other models would utilize such factors as net present value and total life-cycle costs for each of the complete alternative systems with all necessary options installed. In short, there is much room for expansion and improvement over the basic model description presented.

In the context of simple spreadsheet program, the results of each of the individual parameter value benefit/ cost analysis would then only provide additional input to the complete analysis process. Other factors may always provide an overriding reason to select systems contrary to what may show to be the "best" choice strictly by the numbers. Vendor reliability, maintenance costs, etc., and the wealth of other considerations must naturally be combined when making the final decision.

IV. GENERAL MICROCOMPUTER DESCRIPTION

A. BASIC CONCEPTS

The approach taken to describe a microcomputer system is to utilize the concept of levels of abstraction. Levels of abstraction allow a general understanding of a system by beginning the descriptions at the most general level first, and then proceeding to increasing levels of detail. The upper level description is often called the macro or system view, with lower levels the detail or micro views. Levels of abstraction also follow the intuitive concept of top down design. This approach also allows limiting the descriptions to an appropriate level of abstraction without proceeding to levels of detail beyond the necessary scope. The following descriptions are not intended to detail the exact operation of microcomputer components, but are intended to identify some of the aspects to be considered when evaluating among microcomputer alternatives.

There are three main functional units to a single-user, single-task microcomputer system. They are an input device, a system or processing device, and an output device. Many however may consider a storage unit and memory unit as main functional components and describe a microcomputer system as consisting of five main functional units. The choice of three main functional units however better describes current microcomputer systems where the storage and memory units are physically built into the system or processing unit.

The above chosen functional units represent a physical orientation in describing the system. There are however, other ways of describing systems based on logical relationships or internal hardware design. One common method bases descriptions on the architecture of the internal microprocessor and its relationship to connected major components. These methods however are not useful to most users.

An input device normally consists of a keyboard. Most business and commercially oriented microcomputer keyboards have a significant number of extra keys over a standard typewriter layout. These extra keys are necessary to provide additional control signals for the computer system and attached peripheral devices. Some keyboards have added a calculator type numeric keypad at one side of the keyboard that allows easy entry of numerical data. Some keyboards provide a number of "function" keys that implement common command sequences to save repetitive typing chores. Further, some systems allow the function keys to be redefined under program control. Current NAVDAC minimum standards call for the use of a "QWERTY" style keyboard with a shift key and control key.

The keyboard for the IBM Personal Computer is controlled by its own dedicated microprocessor and can therefore be completely controlled by software. This feature allows the entire keyboard to be redefined into any configuration as desired. This feature is particularly useful when used to prevent erroneous data entry. Also, this feature has significant benefit for handicapped persons who can now define their own key configurations and eliminate the need for simultaneous multiple key strokes (such as shift keys for capitals).

There are a number of other input devices supplementing the keyboard device in many current microcomputer systems. These devices consist of touch screens, digitizer tablets, graphics tablets, joysticks, trackballs, lightpens, barcode readers, mouse devices, voice activated input, and remote device communication channel input. All of the above devices have quantifiable parameters for comparison and decision purposes if required.

An output device normally consists of a cathode ray tube (CRT) display screen or video display terminal (VDT), in either monochrome (single color) or color. Microcomputer CRT screens are normally classified as to the amount of character information they can display in text applications. Current NAVDAC standards recommend a minimum of 80 amber or green characters per line by 24 lines on a twelve inch diagonal screen. The new proposed NAVDAC standard calls for a minimum of 80 characters by 24 lines with a 25th line for status information.

Graphics applications normally view the screen as a number of displayable dots or pixels. Each dot is separately addressable, and depending upon the application program and microcomputer system, can be shown in different colors and attributes (such as blinking or intensified). The number of horizontal and vertical displayable pixels defines the screen resolution. A typical microcomputer system might have 320 by 200 pixel resolution. NAVDAC does not presently specify a standard for graphics.

The other major class of output devices is the hard copy printer. Printers for microcomputers are presently of two major types, dot matrix, and fully formed letter quality. Ink jet and laser printers are also just beginning to appear in low cost models. The new proposed NAVDAC standard calls for dot matrix printers to have a minimum print speed of 80 characters per second and letter quality printers to have a minimum 15 characters per second print speed.

Most present microcomputer systems support asynchronous serial communication. NAVDAC recommends that purchased microcomputer systems be able to provide full duplex asynchronous serial communications at speeds from 300 bits/second to 9,600 bits/second. Additionally the system should be able to transfer over standard telephone lines connected to a modem at either 300 bits/second or 1200 bits/second using standard protocols. Since this capability is an absolute requirement and does not vary significantly from system to system, asynchronous communications is not considered in the proposed model.

The proposed model therefore considers a microcomputer system to be composed of a keyboard input device, a system or processing unit which includes memory and a secondary storage device, and a CRT display output device. These three functional units will be quantified to formulate the hardware portion of the proposed model. The major three functional components are also normally purchased together as one unit from the same manufacturer.

The microcomputer operating system will be the final component of the proposed model. Without the operating system and an application program the microcomputer is a useless device. The proposed NAVDAC standard recommends compatibility with the MS-DOS operating system or an operating system that can execute MS-DOS compatible programs.

The next level of abstraction will detail the major components of the microcomputer system and processor unit. The most important component of the entire microcomputer system is the microprocessor. The choice of the microprocessor virtually determines and dominates every other aspect of the entire microcomputer system. The microprocessor is often called the central processing unit or CPU. The CPU through various peripheral and support electronic components or chips enable and interpret input from the keyboard device and formats and processes the output to the CRT device.

The CPU processor must have access to memory to store the list of instructions to operate on and to store results of processing and other outputs. Memory in a microcomputer system is normally composed of chips called dynamic random access memory or DRAM (often abbreviated to RAM) and read only memory or ROM.

The term dynamic refers to the fact that RAM memory is not permanent. RAM memory must be periodically and electronically recharged or refreshed. This means information in RAM memory is only retained while power is available and the microcomputer system is running. Turning the microcomputer off or any interruption in power causes loss of information stored in RAM memory. The term random refers to the fact that each individual storage location can be independently accessed. RAM memory is also often called read/write memory, because new information can be written into and read from the memory. It is commonly used for the storage of user programs and data.

Read only memory however can only be read. ROM memory is still random in that individual storage locations can be independently accessed. ROM memory usually contains the instructions necessary for the microprocessor to automatically bring the microcomputer to a state ready for user input. ROM based routines often contain sophisticated diagnostic and device initialization routines. This process is normally termed "booting" the system and refers to the analogy of pulling oneself up by ones bootstraps automatically without intervention. If the ROM routine necessary to start a system must read information on a secondary storage device this information is appropriately called the bootstrap loader.

Another type of memory found in microcomputer systems is static RAM which does not need to be refreshed but still loses its contents when power is lost. Static RAM is more expensive and currently does not offer the storage density of DRAM. Therefore it is not usually seen in general purpose microcomputer systems.

Memory in microcomputer systems has an access time associated with it. This access time is measured in nanoseconds (ns). A typical microcomputer system has a RAM memory access time of 400 ns or less. Significant improvements in system performance can be seen simply by using faster access memory (often constrained by microprocessor clock speed).

Another form of memory is bubble memory which retains its contents even when power is lost; this is also read/write memory. Bubble memory, however, has significantly slower access times than RAM or ROM and also is not prevalent in microcomputers as main memory. Bubble memory is beginning to be utilized in microcomputer systems, however, as a small size fast access secondary storage device.

In order to save the contents of information stored in RAM, a secondary storage device must be provided. In microcomputer systems this secondary storage usually takes the form of floppy diskettes. The floppy diskette is the medium the information is stored on. The device that reads the medium and transfers information from the floppy diskette to memory is the floppy diskette drive. The diskette drive or drives are normally part of the system unit but can be contained in external cabinets.

Microcomputer systems originally contained disk drives that handled disks that were eight inches in diameter. Current microcomputer systems usually utilize diskette drives that handle diskettes that are five and one quarter inches in diameter. Industry practice distinguishes between the two sizes by the disks (8 inches) versus diskettes (5 1/4 inches) terminology. This verbal distinction is becoming increasingly blurred however as even smaller diskettes enter the marketplace. This next smaller size is three and one half inch. The physical size of the medium however is unimportant to most users. The important characteristic is how much information the medium can store.

The unit of measure for these devices is normally the byte. A byte consists of eight bits of information. Bytes are measured in multiples of powers of two. Two to the tenth power represents 1,024 bytes or one kilobyte (KB). Two to the twentieth power represents 1,048,576 bytes or one megabyte (MB). Kilobytes and megabytes are the standard units to measure capacity of memory and secondary storage devices.

The disk medium can be either soft-sectored or hard-sectored. Soft-sectored means that the sector information is stored under software control. Hard-sectors physically separate and define the sectoring information. The disk medium can also be either single-sided (information on one side only) or double-sided. Further, there can be single, double or quad density which refers to the compactness with which the information is stored by tracks. Single density store approximately 24 tracks of data per inch, double density 48 tracks per inch and quad density 96 tracks per inch. NAVDAC proposed standards call for a minimum of two five-and-one-quarter inch, soft-sectored, dual-sided, dual-density diskettes drives.

Diskette based systems are considered to be medium storage devices in terms of the volume of information stored. If large volumes of information are required hard disk or fixed disk storage devices are used. Hard disks or fixed disks usually are capable of storing thirty times or more the volume of a floppy diskette. Hard disks have significantly faster access times than floppy disk. The proposed NAVDAC standards recommend optional 10 megabyte hard disk when storage requirements exceed that of floppy diskettes.

The CPU or microprocessor is normally connected inside the system unit to main memory consisting of ROM and RAM, to several types of interface chips or controllers, to an input device, and an output device. Two common interface chips or controllers in most microcomputers are the direct memory access (DMA) controller, and the peripheral input/output (PIO) controller.

The DMA controller allows input/output to be placed directly from an input/output device or secondary storage to main memory and vice-versa without going through or interrupting the microprocessor. The PIO controller handles the necessary control and formatting information for communication between the microprocessor and the input/output devices.

The CPU, ROM and RAM, the controllers and input/output devices are connected together by a system bus. The system bus carries the necessary address, data and control signals that allow various components connected to the bus to exchange information. The system bus normally consists of three separate bus structures. They are the address bus, the data bus, and the control bus. Some microcomputer systems may multiplex one or more of these buses on the same physical lines, but most current systems maintain at a least double bus structure.

The address bus carries address signals that identify the separate memory locations and the input/output locations to the CPU. The data bus carries actual data information from the CPU or one component to another. The control bus carries the necessary control and timing signals under the direction of the CPU that coordinates the action of the address bus and the data bus, and carries out the instructions that the CPU is performing. The control bus, for example, would contain the control signals to determine whether a memory access is a read operation or a write operation.

Important aspects from the quantification point of view focuses on the address bus and the data bus. The control bus while being the most important bus, since it contains the control signals, is basically transparent in its operation and not normally available for use by the user or the system programmer. The address bus and the data bus however determine some fundamental aspects and measurable parameters of the microcomputer system.

The address bus and data bus operate on a specific number of bits in a parallel fashion. The specific number of bits that these busses act on at one time is often called the bus width. Thus if the data bus operates on eight bits of information (one byte) in parallel it is said to be an eight bit data bus.

The more information that can be handled at one time, the faster and more efficiently a microcomputer system will operate. For example, if the unsigned number 512 were to be transferred on an eight bit data bus, it would have to occur in two cycles. The number 512 requires a representation of at least nine bits and the first cycle would transfer the first eight bits and another cycle would be necessary to transfer the remaining bit. Intuitively, the larger the data bus, the more data that can be transferred efficiently.

The size of the data bus is often used to determine or name the classification of microcomputer systems. If the data bus is 8 bits wide the microprocessor and microcomputer is termed to be an 8 bit device. If the size of the data bus is 16 bits, a 16 bit microcomputer results. This classification terminology is not universally accepted however.

Similarly, the size of the address bus determines the maximum amount of memory or devices that can be directly accessed by the system. An address bus that is 16 bits wide can access directly sixty-four kilobytes (64KB) of memory (assuming the microcomputer is equipped with at least 64KB of memory). Due to the binary nature of the address bus every additional bit in width doubles the direct addressable memory. An address bus of twenty bits can directly access one megabyte (1MB) of memory.

As application programs become more complex, ever increasing needs for large amounts of directly addressable memory become necessary. In addition, and similar to mainframe computers, multitasking and multiuser microcomputer systems will require relatively large amounts of directly addressable memory.

B. THE MICROPROCESSOR

The last level of abstraction will be the basic internal structure of a typical microprocessor. It must be realized that microprocessors have advanced to a stage where general purpose descriptions no longer adequately portray the many and varied design philosophies. The proposed NAVDAC standard recommends microcomputer systems utilize the Intel 8088/8086 microprocessor or equivalent.

The major functional areas inside a general purpose microprocessor designed for microcomputer systems almost directly parallels the general purpose description of the microcomputer system itself. The CPU must have internal data, address and control busses. It is important to realize and distinguish that the internal width of the CPU busses can be and often are a different size than the external busses. An example of this is a CPU that has an external 8 bit data bus but internal 16 bit data bus. The CPU will fetch two 8 bit bytes sequentially and then can execute on the entire 16 bit value simultaneously.

The CPU busses are normally connected to CPU registers, and input and output buffers. The registers are an array of internal storage locations inside the CPU that hold address, data, and intermediate values awaiting or resulting from computations. Some microprocessors have input/output data and address buffers separate from the register set, while others will utilize the register set itself to act as the necessary buffers. Additionally, some microprocessors dedicate specific registers for specific functions. The most common register-specific function is to have one register dedicated as an accumulator.

A CPU, in addition to the general purpose registers, will have a set of storage locations dedicated to other specific functions. An instruction register contains the instruction that is currently being executed by the microprocessor. A program counter contains the address of the next instruction to be executed. Finally, a status register will maintain certain status information about the state of the microprocessor, or flags that are set or reset depending on results of specific operations.

An important internal component of the CPU is the arithmetic and logical unit (ALU). The ALU performs all calculations, data manipulation, and data comparison functions. This includes executing such operations as comparing values, complementing values, and shifting values left or right. If the CPU has a dedicated accumulator register, the results of operations performed by the ALU are returned or placed into the accumulator.

The control unit within the CPU performs all instructions not directly requiring use of the ALU. This includes operations such as transferring the value of one register to another or transferring the contents of registers to memory or the I/O devices through the system bus. The control unit also controls the operation of the ALU and in conjunction with it, executes the entire set of instructions the CPU is capable of performing.

Within the control unit will be encoded the entire instruction set of the microprocessor. This instruction set is a fixed set of operations that the microprocessor is capable of performing. The instructions are executed on operands that determine the exact nature of the instruction to be executed. For example an add instruction may add values from one register to another, from a register with a value in memory, or from a value in memory to another memory location that points to yet another memory location containing the actual value. The different ways an instruction operand affects the control of the instruction are called addressing modes. Addressing modes extend the capability of an instruction set by allowing different operations to take effect on a single instruction based on the operands of that instruction.

C. MICROCOMPUTER SYSTEM CONSIDERATIONS

There are several important considerations involved when trying to evaluate microcomputer systems. Though the microprocessor utilized is an important system component, there is more to evaluating the microcomputer system than only assessing the hardware capabilities of the microprocessor alone.

The way the system is capable of interacting with various I/O devices is another consideration. It has been shown that systems with DMA offer improved capability and performance over systems where all I/O activity must go through the microprocessor. Additionally, some I/O devices may possess their own processing capability, which frees the main microprocessor from performing what are sometimes called housekeeping functions.

Some microprocessors have the capability to address I/O devices separately from memory addresses. This capability is called I/O mapping and has several advantages. Because an I/O address can be significantly smaller than a memory address, the I/O address can be specified directly without an address calculation. The shorter address also results in shorter I/O commands. The alternative to this is memory mapped I/O, where certain memory addresses are reserved for particular I/O devices and other information they might need. This results in less memory available for application programs. However, the same instructions which are used for memory operations can be used for I/O operations and I/O data transfers do not have to go through an internal register.

Another consideration is whether the system supports interrupts. Interrupts allow the I/O devices to signal the microprocessor only when they need servicing or require an information transfer. Without interrupt capability the microprocessor will have to spend unnecessary time polling I/O devices just to see if servicing is required. Most current microprocessors support interrupts, but how they are implemented and the number supported can have significant impact upon system performance and capability.

Finally, the speed the system operates at is of primary importance. There is often a distinction between the internal speed of the microprocessor and the speed of the rest of the system. Current microprocessors operate in a two to twelve megahertz (MHZ) range. RAM memory on the other hand operates in the 50 to over 450 nanosecond (ns) access time range. A 4 MHZ microprocessor with a 250 ns processor cycle time operating with 200 ns memory attached will obviously exhibit better performance than the same system with 400 ns memory chips attached, because in the former case the processor will not have to wait on memory accesses. Similarly, a 6 MHZ version of the same processor would achieve better performance. These two interdependent times are often critical in their interaction and cannot often be changed once the system design is set.

The speed of operation of a microcomputer system is usually controlled by a crystal oscillator. The oscillator provides an input to the microprocessor, where the resulting timing signals are critical to the operation of the control unit. One must be careful, however, not to base system performance on crystal frequency alone as often the frequency is modified or divided by support chips to provide one or more actual operating frequencies. Also, the actual speed of the processor is not a true indication of system speed, as indicated above with reference to memory access time.

In attempting to provide an accurate but simple quantification method for microcomputer system selection, careful attention must be paid to components or parameters that significantly affect the operation of the entire system. In this context one must be fully aware of the positive and negative interactive aspects of the chosen parameters, and insure that the parameters chosen reflect system capability and performance that are relevant to the user's application and needs.

V. CURRENT 16-BIT MICROPROCESSOR DESCRIPTIONS

A. INTEL 8086/8088 MICROPROCESSORS

The 8086 and 8088 manufactured by Intel Corporation are currently the most widely used 16-bit microprocessors in the microcomputer industry. This is a direct result in the use of the 8088 in the IBM Personal Computer and compatible machines. The 8088 is in such demand that Intel has second sourced its manufacture to both IBM and Commodore Business Machines. The 8086 and 8088 were first commercially introduced in 1978.

The 8086 is a true 16-bit processor having both internal and external 16-bit data paths. The ALU is 16-bits wide and can perform both byte and 16-bit word operations. The address bus of the 8086 is 20-bits wide giving the 8086 an one megabyte physical address range (1,048,576 bytes). The 8086 has 97 basic instruction types which include a hardware multiply and divide instruction and various string manipulation instructions. The 97 instructions work in conjunction with 24 different addressing modes. All instructions are byte oriented with no instruction requiring more than six bytes of operands. Additionally, any instruction can start at any address point.

The 8086 microprocessor is divided into two major parts, an execution unit and a bus interface unit. These two parts are independent and form the implementation of the 8086 pipelined architecture. This means that instructions and data are prefetched by the bus interface unit and placed into a six byte queue or pipeline and then are ready for immediate execution by the execution unit. This results in greatly improved performance characteristics because the bus interface unit can fetch instructions and data simultaneously while the execution unit is executing instructions. The pipeline must then only be cleared and refilled when program execution transfers to a non-sequential address.

The 8086 includes a total of fourteen 16-bit wide registers. There are eight general purpose registers, four of which can be accessed with either 16-bit or 8-bit data. There are four 16-bit segment registers which are used for address calculations. Finally there are two 16-bit control registers one of which is the flags or processor status register and the other the instruction pointer register. There are many 8086 instructions that restrict the operand values to be placed in designated register locations, therefore the register set does not provide completely general purpose registers.

The instruction pointer register is normally called the program counter in other microprocessor implementations. Intel refers to this register as the instruction pointer because this 16-bit quantity is combined with one of the 16-bit quantities in the segment registers to form the 20-bit address. The appropriate 16-bit segment register is shifted left four bits and added to the instruction pointer to form the address. Another reason for the change in terminology is that the 8086 instruction pointer points to the next instruction to be fetched and not the next instruction for execution (which is in the six byte pipeline).

In addition to supporting the one megabyte physical address range the 8086 can address 64 kilobytes of I/O ports. The 8086 instruction set has special input and output instructions to send and receive data from the I/O ports specified by the port address. The processor and address bus distinguish between memory address and I/O address by special control signals. The use of separately addressed I/O ports allows the 8086 to use both memory mapped and I/O mapped devices.

An example of both memory and I/O mapping occurs in the IBM Personal Computer where the display screen is memory mapped but the cathode ray tube (CRT) control circuitry has its own processor controller with additional registers that can be changed via the I/O ports. This results in the display screen attributes and cursor controls to be under programmer or application program control. I/O mapping also saves memory space in that device control information does not have to be stored or utilize a portion of system memory.

Another important characteristic of the 8086 processor is its interrupt handling capability. The 8086 processor is designed to utilize both hardware and software interrupts. A total of 256 different interrupts are supported. The first 1024 bytes of memory in 8086 systems are dedicated to hold an interrupt vector table. When either a hardware or software interrupt occurs, the interrupt instruction retrieves an address pointer in the interrupt vector table and branches to that address to perform the interrupt service routine. This allows all interrupt service routines to be written or changed by the user. This can be done either by replacing the existing routine with another or changing the address of the interrupt vector pointer to a new location that contains the desired service routine.

The 8086 processor was originally designed to be driven by a 5 MHZ clock for timing and control purposes. There are now 8 MHZ and 10 MHZ versions available.

The 8086 design allows for coupling with other 8086 processors for parallel execution or it can utilize two special coprocessors. The 8087 Numeric Processor Extension (8087 NPX) can be utilized to implement full 80-bit IEEE floating point arithmetic operations by adding an additional register set of eight 80-bit registers and 68 new floating point arithmetic instructions. The 8089 Input/Output Processor (8089 IOP) is a separate microprocessor with its own instruction set that can be linked with 8086 systems for high speed I/O control of systems with numerous I/O devices. The 8089 in general acts as a sophisticated and high speed I/O and DMA controller.

The 8088 is functionally equivalent to the 8086. The only difference is that the 8088 has an 8-bit external data bus vice 16-bit and the instruction pipeline is four bytes long versus six. Internally the 8088 processes data and instructions identically to the 8086 and all programs and applications developed for 8086 systems will run on 8088 systems and vice-versa.

Intel Corporation is providing a family of processors and coprocessors beginning with the 8088/8086 that are upward compatible. The next set of processors in the series is the iAPX 186/188 (80186/80188). These two processors are functionally equivalent to the 8088/8086 but additionally have several peripheral support chips designed into the microprocessor itself. The 80186/80188 series combine clock generation, two independent DMA channels, programmable interrupt controller, three programmable timer/counters, programmable memory-select and peripheral-select logic, a programmable wait-state generator, and a local bus controller Systems designed with iAPX 186/188 need approximately twenty fewer chips which simplifies the circuit board design and reduces cost.

The chips also have improved internal design resulting in significantly faster execution for many instructions. Additionally, ten new instructions have been added for more efficient coding of higher level languages. Address calculations use a dedicated hardware adder which significantly increases the speed at which the bus interface unit can fill the execution unit pipeline. Both chips also have newly designed coprocessors the iAPX 187 and 189.

Another microprocessor in the Intel family is the iAPX 286 (80286). This is a greatly improved 8086 model supporting one gigabyte of virtual memory and up to 16 megabytes of physical memory for use in multiuser and multitasking applications. All memory management functions are built into the chip. In development is the Intel iAPX 386 which will be a full 32-bit microprocessor and still planning to be upward compatible with applications developed for 8086 systems.

B. ZILOG Z8000 MICROPROCESSOR

The Zilog Z8000 was introduced in 1979 and is available from Zilog Corp., and is also second sourced from Advanced Micro Devices (AMD). Z8000 is the name for a family of processors. The Z80 8-bit microprocessor is not upward compatible with the Z8000 as the Z8000 employs a totally new and different architecture. Zilog claims however that conversion of Z80 programs to the Z8000 is relatively straightforward.

The Z8000 is also a true 16-bit microprocessor supporting 16-bit data paths both internally and externally. Additionally the Z8000 instructions are either 16-bits or 32-bits wide with the instructions mostly word oriented (16-bits). The ALU is 16-bits wide and supports individual bit set and clear operations in addition to byte and word data types. The Z8000 can directly address eight megabytes of memory. The eight megabytes of directly addressable physical memory is divided into 128 segments of 64 kilobytes each. The 128 segments is provided by an additional seven bits or address lines giving the Z8000 twenty-three address lines (verses 20 for the 8086).

The Z8000 provides 110 basic instructions implemented by random logic. The instruction set can operate on 8, 16 or

32 bit quantities and instructions vary in length from two to eight bytes. The instructions however are required to be word aligned, which means they must begin on even byte addresses and must have an even number of bytes. The Z8000 has eight different addressing modes.

The Z8000 implements a single word pipeline. Only the next single word is prefetched from memory into an instruction pipe. This can only occur during the current instruction execution and only if the current instruction does not require the use of the address bus. Although this is not as a sophisticated instruction pipeline as the 8086, another element of sophistication is added by allowing the prefetched instruction to be decoded in the buffer. Zilog calls the hardware portion of the processor that performs this a Lookahead Instruction Decoder and Accelerator.

The Z8000 contains a total of twenty-two 16-bit registers. There are sixteen general purpose registers. Of these, all but one of the registers can be used either as accumulator, index pointer or memory pointers. Thus the Z8000 does provide true general purpose registers. In addition the first eight registers can be accessed with either byte or word values, and the first sixteen registers can be combined to form either eight 32-bit registers or four 64-bit registers. This allows the Z8000 to easily perform 16-bit and 32-bit multiplies and 32-bit and 64-bit divides. This also allows relatively faster string manipulation instructions.

The remaining registers are composed of two registers for implied system and normal stack pointers, one program counter register (PC), one program status area pointer (PSAP) register, which contains the segment number and offset, one flag and control word register (FCW), and a refresh counter register.

The Z8000 like the 8086 supports separate I/O addressing and also supports 64 kilobytes of I/O ports. Additionally the Z8000 has a special system mode of operation that allows the addressing of an additional 64 kilobytes of I/O space. This additional I/O space will not be considered for the single user evaluation, but should be added for multi-user and multi-process implementations.

The Z8000 supports 256 different interrupt types and like the 8086 the interrupt pointer tables are easily modified. The pointer tables are not stored in the lowest segment of memory however, but rather at a location pointed to by the PSAP register upon system initialization. The Z8000 is currently available in 4 and 6 MHZ versions.

Z8000 is actually the general designation of a family of Zilog microprocessors. The above descriptions apply to the actual Z8002. Zilog offers the Z8001 that provides a couple of extra registers that when combined with the Z8010 Memory Management Unit (MMU) extends the addressing range of the Z8000 to 48 megabytes of physical memory. This is accomplished by managing six segments of eight megabytes each. The MMU also allows randomly relocating the 128 segments (64 kilobytes each) in any of the six larger segments.

C. THE MOTOROLA MC68000 MICROPROCESSOR

The MC68000 was commercially introduced by Motorola Semiconductor Division in 1980. The MC68000 due to its later release has gained a technological sophistication beyond the Intel 8086 and Zilog Z8000. The MC68000 contains a very complex internal architecture with over three times as many transistors implemented in silicon over the 8086 and Z8000.

The MC68000 is more than just a 16-bit microprocessor. The MC68000 has an external 16-bit data bus but is internally 32-bits wide for all data and address registers. Therefore depending on how classification of the processor is done the MC68000 microprocessor could be considered a 32-bit microprocessor. The ALU however is only 16 bits wide and can operate on bit, byte, 16-bit word and 32-bit double word basic data types.

The MC68000 is packaged in a 64-pin chip and implements a separate 16-bit data bus and a full external 23-bit address bus. The address bus is not multiplexed with the data bus as in the 8086 and Z8000. The 23-bit external address pins are used in conjunction with a 24-bit internal program counter. This implementation allows the MC68000 to have a completely linear sixteen megabytes of physical address space. The address space is completely linear in that there are no segments that break the address space into separately addressable blocks. All addresses are specified by the full 24-bit value.

The internal design of the address bus is 32-bits wide. The external 23 address lines were limited due to packaging limits only. Therefore the MC68000 has the capability to expand to a full 32 bits of address supporting four gigabytes of physical memory merely by changing the chip packaging. No redesign of the internal chip architecture is required.

The bottleneck in the MC68000 restricting it from being a full 32-bit microprocessor internally is the 16-bit wide ALU. All thirty-two bit operations from the 32-bit registers must be accomplished in two consecutive cycles through the 16-bit ALU.

The MC68000 microprocessor therefore is basically a 16-bit device with significant extensions. The architecture as implemented is not as symmetric (32-bit data and address bus internally, only 24-bits address bus currently used, 23-bits of address bus externally, 16-bit external data bus and 16-bit internal ALU) as the Z8000 or the 8086, but offers a large physical address space and increased expansion opportunity.

The MC68000 has only 61 basic instructions which are implemented in microcode. Instructions vary in length from two to ten bytes. The low number of instructions is somewhat misleading in that there are a number of high level instructions that would require the use of three or four instructions on other processors. This advantage is offset by the fact that the MC68000 contains no specific string manipulation instructions. More combinations of MC68000 instructions and programming effort are required to implement word processing functions. Instruction execution on the MC68000 requires that instructions be word aligned

The MC68000 implements a single prefetched instruction pipeline similar to the Z8000. The prefetch mechanism contains enough information to execute current instruction, decode next executable instruction, and prefetch the following instruction from memory simultaneously. The mechanism also attempts to predict the most likely address for branch instructions.

The MC68000 contains a total of 19 registers. All of the registers except the status register are 32-bits wide, with the status register 16-bits wide. The register set contains eight completely general purpose data registers, seven address registers, two system stack pointers, a program counter register, and the status register. The two system stack pointers are designated as a user stack pointer and supervisory stack pointer.

All of the data registers can be accessed as either byte, word or double word values. All addressing modes and instructions can use any of the eight general purpose registers for operands. The eight address registers allow easy implementation of multiple stacks, data, and program areas.

The MC68000 is completely memory mapped and offers no separate address space for I/O devices or ports. This is not considered a restriction because of the large address space available. This does mean that in a system implementation certain areas of memory must be reserved for I/O control information. The programmer must then insure that application programs do not interfere with the designated areas. With the linear address space of the MC68000 keeping data and program areas separated is totally the responsibility of the programmer. There are no segments to allow the placing of various program parts (program, data, stack, I/O) automatically into separate areas.

The MC68000 supports 256 vectored interrupts virtually identical to the 8086 method. A interrupt vector table is located at the lowest memory addresses. Unlike the 8086 and the Z8000 however, the MC68000 starts at the lowest address to begin execution upon initial power on sequence. This means some initial interrupt pointers must be available and fixed in a system ROM. Once loaded the interrupt vector table is as easily changed and modified as the 8086 and Z8000. The MC68000 is now available in 8,10, and 12.5 MHZ versions.

The MC68000 has a series of coprocessors available to extend the capability of the basic processor. There are also several versions of the basic MC68000. There is a MC68008 similar in concept to the 8088 that is code compatible with the MC68000 but has an 8-bit data bus and utilizes only 20-bits of address bus for one megabyte memory systems. The MC68010 is an enhanced MC68000 that supports virtual memory and when combined with the MC68451 Memory Management Unit can additionally support up to 64 MB of address space and sophisticated user and supervisor modes. Finally Motorola is testing the MC68020 which is a full 32-bit microprocessor with 32-bit data, address, and ALU capability.

VI. EXAMPLE 16-BIT COMMERCIAL OPERATING SYSTEMS

A. DIGITAL RESEARCH CP/M-86

Control Program for Microcomputers (CP/M) was first developed for an Intel 8080 microcomputer system. CP/M was later adapted to 8085 and Z80 microcomputer systems and became the de facto 8-bit operating system standard. NAVDAC recommended that all 8-bit microcomputer systems for general Navy use be compatible with CP/M Version 2.2. There are literally thousands of serious business applications designed to run under CP/M and all CP/M systems can exchange data with standard 8 inch disk formats.

CP/M-86 is the version of CP/M developed for 16-bit microcomputer systems using the 8086/8088 microprocessor. It maintains virtually the same user interface and command syntax. Applications developed to run under CP/M 2.2 will not run under CP/M-86 without considerable conversion.

CP/M-86 is composed of three major parts. They are the Basic Input/Output Operating System (BIOS), the Basic Disk Operating System (BDOS), and the Console Command Processor (CCP). CP/M-86 requires a minimum of 57 kilobytes of memory in order to execute user programs. The CP/M-86 program itself when loaded into an user machine occupies approximately 13.5 kilobytes of RAM and the remaining RAM in the user system is designated as the Transient Program Area (TPA). Although CP/M-86 normally resides in the lowest segment of memory after the interrupt vector table, it is designed to be relocated in any location as necessary.

CP/M-86 is designed such that the BDOS and CCP present a common user interface in all CP/M-86 implementations. This means that the BDOS and CCP are the same and implement their functions through calls to the BIOS. The BIOS is the device dependent portion of the operating system and must be specifically tailored or customized for each different microcomputer system. This design allows CP/M-86 to appear device independent to the user or application program. Also, due to the similarity of the BDOS and CCP in all CP/M versions, users can immediately begin to use the CP/M operating systems on a wide variety of microcomputer systems.

The BDOS is the nucleus of the CP/M-86 operating system providing the necessary routines to manage the secondary storage medium and act on commands passed to it from the CCP. CP/M-86 (Version 1.1) provides 49 different BDOS functions. This includes 17 additional functions over the CP/M 2.2 8-bit functions. These functions provide for such elementary operations as open file, close file, rename file, read random record, and basic user console functions such as read a character from the keyboard and display a character on the CRT screen. All CP/M-86 BDOS function calls are implemented by placing the function number and other information into designated registers of the 8086/8088 processor, and then issuing a reserved interrupt request.

The BIOS of CP/M-86 provides 21 functions for the applications programmer. Because some BDOS functions pass through the BIOS, some of the BIOS functions duplicate functions available in the BDOS. The two most notable examples are character input/output from the keyboard to the display screen. The BIOS does not duplicate the BDOS exactly because the BIOS contains only enough information to handle the specific hardware functions of the system machine, while the BDOS provides many routines that do not function directly with system hardware. Some other examples of specific BIOS functions are select disk drive, track, and sector, or read selected track and sector. Thus the BIOS provides those functions necessary to deal with physical facets of the user machine while the BDOS acts on logical aspects of applications programs. In total CP/M-86 provides 70 functions for the applications programmer.

The Console Command Processor is responsible for interpreting and acting on all user commands typed from the keyboard. The CCP receives input from the keyboard and parses the input to determine if the input is an operating system command, a command to load and execute an application program, or a command CP/M-86 does not understand that must be handled by an error routine. The CCP presents the user interface for the particular microcomputer system.

The CCP has two major parts. The first part contains those commands and their associated routines that CP/M-86 operates on directly and are internal to the CCP. These routines and commands are immediately available to the user when ever the operating system is loaded and active. CP/M-86 calls these "Built-In Commands". CP/M-86 contains six such commands. There are two types of directory commands which list the names of either system or user files on the secondary storage device, two commands to either erase or rename file names, a command to list (type) the contents of a file, and a command to change the active user number. CP/M-86 allows the designation of 16 user numbers for primitive file protection purposes.

The second part of the CCP takes action when the current command is not one of the built in commands. In this case the CCP assumes that the command is a directive to load and execute a program contained on a secondary storage device. The CCP then actives the device and searches for the given program. If found, the CCP performs the necessary initialization routines to load the program into memory and pass control to it for execution.

CP/M-86 provides thirteen external programs and calls them "Transient Utility Commands". These programs provide status information for secondary storage and routines to transfer files from one device to another. There are several external programs that provide functions not found in many microcomputer operating systems.

These functions include a help program which provides an on-line help facility for all CP/M-86 commands and their correct syntax. This is a significant contribution to user friendliness for a microcomputer operating system. An 8086/8088 assembler and debugger are also provided as part of the operating system. These two programs can cost hundreds of dollars for particular microcomputer systems when they are not provided. CP/M-86 also provides a limited line oriented editor for program development.

In the CP/M-86 implementation specifically for the IBM Personal Computer, two additional external commands are provided. One is a menu driven setup program that uses function keys to easily set and assign system parameters, such as communication port initializations, printer selections and other functions. The second command is a print utility that provides for limited virtual spooling of output for an attached printer device.

Current versions of CP/M-86 can support up to sixteen attached floppy disk drives. The transient utility command "DISKMAINT" (for disk maintenance) is the CP/M-86 routine that sets up and initializes floppy disks to the CP/M-86 format. This format will support either single-sided diskettes or double-sided diskettes. The format routine in DISKMAINT will format 40 tracks per side creating either 156 kilobytes or 316 kilobytes of secondary storage capacity respectively. CP/M-86 divides a disk into thirty-two physical sectors and records 128 bytes per sector, reserving one track of information for internal use. CP/M-86 will allow a total of 64 directory entries (files) per diskette.

No support is currently provided for hard disk drives. However, most hard disk manufacturers will provide a customized version of the CP/M-86 BIOS to support their particular hard disk. The file structure implementation and storage capacities will then depend on the hard disk manufacturer.

B. MICROSOFT MS-DOS AND IBM PC-DOS

Microsoft Corp. developed the MS-DOS (Microsoft Disk Operating System) operating system under contract by IBM specifically for the IBM Personal Computer. As such when IBM released the Personal Computer the operating system was simply named DOS (Disk Operating System). Because there are some minor differences[2] between the version that Microsoft now markets independently and the version for the IBM Personal Computer, the Personal Computer version is generally referred to as PC-DOS.

When Microsoft developed PC-DOS the CP/M operating system for 8-bit microcomputers was considered the de facto standard. Microsoft wanted to maintain some CP/M compatibility so that it would be easy for developers to convert programs for the new system, but also felt that CP/M could be improved in a number of areas. The result is that PC-DOS and CP/M or CP/M-86 share similar functional structure but differ on exact details of implementation and command syntax.

---

[2] MS-DOS and PC-DOS are exactly identical version for version to the applications programmer. All internal functions and operating system calls are precisely identical in operation. The only differences are some command names that are slightly changed and the addition or deletion of some external program utilities depending on a particular vendor. Most systems utilizing MS-DOS include an object module library utility that is not included in PC-DOS. PC-DOS allows the linking of separate object modules or libraries but does not include the utility to add object modules to an existing library.

Like CP/M, MS-DOS is composed of three major parts, and these parts are equivalent in purpose. IBM names these parts the IBMBIO (IBM Basic Input/Output System), IBMDOS (IBM Disk Operating System), and the COMMAND Processor (Microsoft names these parts I/O System, MDOS, and COMMAND).

The IBMBIO contains the device specific information necessary to communicate with the physical devices attached to the IBM Personal Computer. In the MS-DOS version this also contains the necessary device information for the particular machine implementation. CP/M-86 builds its own set of jump vectors that perform similar to an interrupt vector table. CP/M-86 BIOS functions are then implemented through this set of jump vectors. MS-DOS on the other hand implements all operating system internal functions through the vector interrupt table of the 8088/8086 microprocessor.

MS-DOS reserves 32 of the 8088/8086 interrupts for its own use. One of the interrupts is reserved for operating system function calls that act in the same manner as CP/M-86 BDOS functions. The similarity between MS-DOS and CP/M-86 is shown by the use of the same function numbers for a large majority of the operating system calls. Because MS-DOS is totally interrupt driven, the I/O system remains more transparent to the programmer than CP/M-86. As implemented MS-DOS makes no distinction between the I/O system calls and the disk operating system calls. MS-DOS (Version 2.0) provides a total of 120 operating system functions of which 83 can be considered user functions (MS-DOS reserves 24 interrupts and 13 interrupt driven function calls for its own internal use).

The MS-DOS COMMAND processor has been significantly expanded and simplified over CP/M-86. The major driving factor for the current expansion is that the latest version of MS-DOS provides internal support for hard disk drives. MS-DOS contains 26 internal commands and 18 external commands or operating system utilities. MS-DOS has incorporated more of the common operating system functions as internal commands and attempted to simplify the command syntax of these commands.

An example of this is the method of copying files from one device to another. CP/M-86 provides a transient utility command named PIP (Peripheral Interchange Program) to accomplish this task. The syntax for use of the command is:

```
PIP device:destination-file = device:source-file options
```

MS-DOS provides an internal command named COPY which the syntax is:

```
COPY device:source-file device:destination-file options
```

From this example the similarity between the two operating systems can be seen. In general however, MS-DOS provides a simpler and easier understood user interface by employing more English-like commands than CP/M-86. Current versions of MS-DOS provide no help facility however.

Another difference is that MS-DOS does not provide its own assembler. It is offered as a separate product from Microsoft. Microsoft normally substitutes its own BASIC language interpreter with the operating system product. MS-DOS does provide a simple line oriented editor (EDLIN) and an 8088/8086 debugger (DEBUG). The latest version of DEBUG (2.0) will assemble 8088/8086 and 8087 mnemonics but offers no macro or named address facility.

The major difference between CP/M-86 and MS-DOS occurs in the manner that MS-DOS handles the secondary storage medium. MS-DOS can support up to 256 attached drives and has built in support for hard disk drives. MS-DOS for floppy disk drives will support either single-sided or double-sided medium. MS-DOS formats the floppy disk medium at 40 tracks per side and either eight or nine sectors of 512 bytes each per side. This gives MS-DOS either 160, 180, 320, or 360 kilobyte drive capacities. MS-DOS does not reserve a dedicated track for its own use unless requested. This allows more information to be stored on data disks as the operating system itself is not stored on the disk.

Normally MS-DOS will allow 64 files per disk for single-sided disk or 112 files per disk for double-sided disks. MS-DOS (Version 2.0) also supports a tree structured file system that allows the creation of separate directories and sub-directories. This was implemented as part of the support for hard disk drives. The result of the tree structured directories is that MS-DOS can support a virtually unlimited number of files. The number of files is limited only by the physical size of the storage medium and is not restricted by the operating system.

It is difficult to compare operating system features at any given point in time because the manufacturers are constantly updating them. The above comparisons discuss two relatively different versions of the two manufacturers operating systems. The prospective Navy purchaser should realize that different versions of the same operating system can have significantly different features and capability, and should become cognizant of the differences before considering a purchase.

VII. CONCLUSIONS AND RECOMMENDATIONS

There are currently so many options facing a prospective Navy microcomputer purchaser, that to decide upon one system among the hundreds that are available certainly seems an impossible task. Standards and guidelines developed by NARDAC Norfolk, under the direction of NAVDAC, go along way to eliminate some of the confusion and lead the prospective purchaser in a forward direction. NAVDAC Pub. 15, Economic Analysis Procedures for ADP, provides guidance on how to conduct economic analysis. The procedures described by Pub. 15 can be utilized to conduct analysis between the many differing microcomputer alternative systems. It is recommended however, that when conducting cost benefit analysis among many differing systems, that not only the absolute cost benefit values be examined, but in addition, the changes in benefits versus the changes in costs, be examined for the different alternatives under consideration. It is recommended that NAVDAC Pub. 15 be updated to reflect this change.

A framework for the construction of a model to perform the change analysis has been suggested using the functions available in common spreadsheet type programs. It is further suggested that these programs provide a means to store and easily manage the now large amount of data necessary in evaluating among many microcomputer systems. It is hoped that the use of these automated tools will provide a positive incentive to perform a more complete and thorough economic analysis. It is recommended that further research be done, particularly in the area of decision support systems and the construction and integration of automated economic analysis models not only for evaluation of microcomputer systems, but to be implemented on microcomputer systems.

It is recommended that NARDAC Norfolk, having the microcomputer systems and cost data available, be a starting point in the construction of some initial working models. Further, since NARDAC Norfolk will begin teaching instructional classes on the use of microcomputer spreadsheets, that these classes will provide an excellent opportunity for the construction of a working microcomputer decision support model.

LIST OF REFERENCES

1. "Personal Computers: and the Winner is IBM.", *Business Week*, pp. 76-96, October 3, 1983.

2. DATA SOURCES *Hardware-Data Communications/1st Quarter 1984*, pp. A50-A77, Ziff-Davis, 1984.

3. "Computer shock hits the office.", *Business Week*, pp. 46-53, August 8, 1983.

4. Naval Data Automation Command, *Microcomputer Support* NAVDAC Advisory Bulletin No. 28.

5. Naval Data Automation Command, *Microcomputer Hardware and Software Standards*, NAVDAC PUB NO. 17.7.

6. Naval Data Automation Command, *Navy-wide Microcomputer Contract*, NAVDAC Advisory Bulletin No. 54, October 27, 1983.

7. Gore, Marvin and Stubbe, John. *Elements of Systems Analysis*, pp. 24-29, Wm. C. Brown, 1983.

8. Zimmerman, Denise C., *Economic Analysis Procedures For ADP*, Naval Data Automation Command, NAVDAC Pub 15, March 1980.

9. Burr, William E., and Gordon, Robert. "Selecting a Military Computer Architecture," *Computer*, October 1977.

10. Fuller Samuel H., and Burr William E., "Measurement and Evaluation of Alternative Computer Architectures," *Computer*, October 1977.

BIBLIOGRAPHY

Allison, Dennis R., "A Comparison of CP/M and UNIX," *IEEE 1982 Compcon Spring*, 1982.

Appalaraju, Ramakanth. "High End uPs Challenge Mainframes," *Digital Design*, October 1983.

Ball, Robert H., *Shipboard Implementation of Microcomputer Systems*, M. S. Thesis, Naval Postgraduate School, June, 1977.

Barden, William Jr., *How to buy and use minicomputers and microcomputers*, Howard Sams & Co., 1976.

Boonham, J.C., ed., *Small Systems Computer Sourcebook*, The Rococo Press Limited, 1978.

Boyce, Jefferson C., *Microprocessor and Microcomputer Basics*, Prentice-Hall, 1979.

Boyle, Brian. "Software Performance Evaluation," *Byte*, February 1984.

Burr, William E., and Gordon, Robert. "Selecting a Military Computer Architecture," *Computer*, October 1977.

Carroll, John M., and Rosson, Mary Beth. "Beyond MIPS: Performance is Not Quality," *Byte*, February 1984.

"Computer shock hits the office.", *Business Week*, August 8, 1983.

Cooper, James Arlin. *Microprocessor Background For Management Personnel*, Prentice-Hall, 1981.

Dahmke, Mark. *Microcomputer Operating Systems*, Byte Publications, Inc., 1982.

Dahmke, Mark. *The BYTE Guide to CP/M-86*, McGraw-Hill, 1984.

DATA SOURCES *Hardware-Data Communications/1st Quarter 1984*, Ziff-Davis, 1984.

Deitel, Harvey M., *An Introduction to Operating Systems*, Addison-Wesley, 1983.

Dietz, William B., and Szewerenko, Leland. "Architectural Efficiency Measures: An Overview of Three Studies," *Computer*, April 1979.

Digital Research, *CP/M-86 Operating System, Version 1.1*, 1983.

Doty, Keith L., *Fundamental Principles of Microcomputer Architecture*, Matrix Publishers, 1979.

Freedman, David. "Portable Operating Systems Fight for 16-Bit Machines," *Mini-Micro Systems*, September 1982.

Friedman, Roy and Legg Gary. "Microcomputer Operating Systems Compete in Portability and Performance," *Mini-Micro Systems*, January 1984.

Fuller Samuel H., and Burr William E., "Measurement and Evaluation of Alternative Computer Architectures," *Computer*, October 1977.

Giuliano, Vincent E., "The Mechanization of Office Work," *Scientific American*, September 1982.

Gore, Marvin and Stubbe, John. *Elements of Systems Analysis*, Wm. C. Brown, 1983.

Greenberg, Robert B., "The UNIX Operating System and the Xenix Standard Operating Environment." *Byte*, June 1981.

Gupta, Amar and Toong, Hoo-min. ed., *Advanced Microprocessors*, IEEE Press, 1983.

Hamacher, Carl V.; Vranesic, Zvonko G. and Zaky, Safwat G., *Computer Organization*, McGraw-Hill, 1978.

Horwitt, Elizabeth and Smith, Amy E., "The Invisible Office Manager," *Business Computer Systems*, March 1983.

Houston, Jerry. "Don't Bench Me In," *Byte*, February 1984.

Intel Corporation. *iAPX 88 Book*, 1981.

Kildall, Gary. "CP/M: A family of 8- and 16-Bit Operating Systems," *Byte*, June 1981

Leedy, Glenn. "The National Semiconductor NS16000 Microprocessor Family," *Byte* April 1983.

Lenk, John D., *Handbook of Microprocessors, Microcomputers, and Minicomputers*, Prentice-Hall, 1979.

Libuda, Paul H., *Personal Computing*, M. S. Thesis, Naval Postgraduate School, Sept., 1977.

Kaisler, Stephen H., *The Design of Operating Systems For Small Computer Systems*, John Wiley & Sons, 1983.

Madnick, Stuart E., and Donovon, John J., *Operating Systems*, McGraw-Hill, 1974.

"Microprocessor Data Manual III," *Electronic Design*, Vol. 27, No. 24.

Microsoft, Inc., *Disk Operating System, Version 2.0*, International Business Machines, 1983.

Naval Data Automation Command. *Microcomputer Hardware and Software Standards*, NAVDAC PUB NO. 17.7.

Osborne, Adam. *An Introduction to Microcomputers, Vol. 0, The Beginner's Book*, Osborne & Associates, 1979.

Osborne, Adam. *An Introduction to Microcomputers, Vol. 1, Basic Concepts, 2nd Ed.*, Osborne/McGraw-Hill, 1980.

"Personal Computers: and the Winner is IBM," *Business Week*, October 3, 1983.

Rector, Russell and Alexy, George. *The 8086 Book*, Osborne/McGraw-Hill, 1980.

Schindler, Max. "Special Report: Check Operating System Features to See How Powerful a uC Will Be," *Electronic Design*, Vol 26, No. 24.

Schmitt, Stephen. "Virtual Memory for Microcomputers," *Byte*, April 1983

Simington, R. B., "The Intel 8087 Numerics Processor Extension," *Byte*, April 1983.

Slater, Leland W., ETC, USN, *Everything You Ever Wanted To Know About Microcomputers* Naval Regional Data Automation Center, Norfolk, June, 1982.

Smith, Donald L., *Method to Evaluate Microcomputers For Non-Tactical Shipboard Use*, M. S. Thesis, Naval Postgraduate School, Sept., 1979.

Sommerville, I., *Software Engineering*, Addison-Wesley, 1982.

Starnes, Thomas W., "Design Philosophy Behind Motorola's MC68000, Part 1," *Byte* April 1983.

Starnes, Thomas W., "Design Philosophy Behind Motorola's MC68000, Part 2," *Byte* May 1983.

Stone, Harold S., "Life-Cycle Cost Analysis of Instruction-Set Architecture Standardization for Military Computer Systems," *Computer*, April 1979.

Swanson, Paul. *Microcomputer Disk Techniques*, Byte/McGraw Hill, 1982.

Taylor, Roger and Lemmons, Phil. "Upward Migration, Part 1: Translators," *Byte*, June 1982.

Taylor, Roger and Lemmons, Phil. "Upward Migration, Part 2: A Comparison of CP/M-86 and MS-DOS," *Byte*, July 1982.

Toong, Hoo-min D., and Gupta, Amar. "An Architectural Comparison of Contemporary 16-Bit Microprocessors," *IEEE Micro*, May 1981.

Wong, William G., "MS-DOS: An Overview, Part 1," *Microsystems*, March 1984.

Wong, William G., "The New 16-Bit Super Microcomputers: A comparative look at the Intel 80286, Motorola 68000, and National 16032," *Microsystems*, November 1983.

Zarrella, John. *Operating Systems Concepts and Principles*, Microcomputer Applications, 1979.

Zarrella, John, ed., *Microprocessor Operating Systems*, Microcomputer Applications, 1981.

Zimmerman, Denise C., *Economic Analysis Procedures For ADP*, Naval Data Automation Command, NAVDAC Pub 15, March 1980.

Zingale, Tony. "Intel's 80186: A 16-Bit Computer on a Chip," *Byte*, April 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Naval Data Automation Command Code 30 Washington Navy Yard Washington, D.C. 20374	2
4.	Naval Regional Data Automation Center Code 31 Norfolk, Virginia 23511	1
5.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
6.	Professor Norman F. Schneidewind, Code 54Ss Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
7.	Computer Technology Programs Code 37 Naval Postgraduate School Monterey, California 93943	1
8.	Lt. Keith G. Highfill, USN P.O. Box 1110 Newport, Rhode Island 02840	2

333-[illegible]

208381

Thesis
H52822 Highfill
c.1 A decision model for
selection of microcomputers and operating
systems.

26 SEP 86 33401

208381

Thesis
H52822 Highfill
c.1 A decision model for
selection of microcomputers and operating
systems.

AMERICAN BINDERIES 4

thesH52822
A decision model for selection of microc
3 2768 002 05988 3
DUDLEY KNOX LIBRARY